Express Mail Label No. EV303486355US

METHOD AND SYSTEM FOR HOSTING AN APPLICATION WITH A FACADE SERVER

BACKGROUND

[0001] Some software applications may utilize a computer network, such as the Internet, to exchange application data with a user. The network may comprise a plurality of devices, such as switches, routers, and computer systems, that are coupled together via a network topology, such as Ethernet and Token Ring. A web-server may establish network connections that reliably transfer the application data across the network. The network connections may facilitate the transfer of application data through one or more network protocols, such as transmission control protocol/internet protocol (TCP/IP).

[0002] When an application and a user of the application are operating on the same computer system, it may be undesirable to utilize a web-server. The web-server may open computer ports and increase the vulnerability of the computer system that operates the application to computer-based attacks, such as denials of service and buffer overflows. In addition, the web-server may require additional processing that may be unnecessary, such as instantiating the TCP/IP stack.

SUMMARY

[0003] One exemplary embodiment may be a system that comprises a central processing unit (CPU), an application, and a facade server. The facade server may host the application without utilizing network protocols. A program executing on the CPU may create an interface between the facade server and a browser for exchanging data associated with the application.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] For a detailed description of the embodiments of the invention, reference will now be made to the accompanying drawings in which:

[0005] Figure 1 illustrates a computer system constructed in accordance with embodiments of the invention;

[0006] Figure 2 illustrates an exemplary data flow between components of Figure 1 in accordance with embodiments of the invention; and

[0007] Figure 3 illustrates a computer system that operates in two modes constructed in accordance with embodiments of the invention.

NOTATION AND NOMENCLATURE

[0008] Certain terms are used throughout the following description and claims to refer to particular system components. As one skilled in the art will appreciate, computer companies may refer to a component by different names. This document does not intend to distinguish between components that differ in name but not function.

[0009] In the following discussion and in the claims, the terms "including" and "comprising" are used in an open-ended fashion, and thus should be interpreted to mean "including, but not limited to ..." Also, the verb "couple" or "couples" is intended to mean either an indirect or direct connection. Thus, if a first device couples to a second device, that connection may be through a direct connection, or through an indirect connection via other devices and connections.

DETAILED DESCRIPTION

[0010] The following discussion is directed to various embodiments of the invention. The embodiments disclosed should not be interpreted, or otherwise used, as limiting the scope of the disclosure unless otherwise specified. In addition, one skilled in the art will understand that the following description has broad application, and the discussion of any embodiment is meant only to be exemplary of that embodiment, and not intended to intimate that the scope of the disclosure is limited to that embodiment.

[0011] Figure 1 illustrates a computer system constructed in accordance with embodiments of the invention. System 100 may be any type of computer system, such as a laptop computer, a personal computer, or a stand-alone computer operated as a server. The system may comprise a single central processing unit (CPU) 102, as illustrated in Figure 1, or may comprise a plurality of CPUs arranged in a configuration where parallel computing may take place. The CPU 102 may couple to a memory 104 that may store a browser 106 program, a plugin 108 program, a facade server 110 program, and an application 112. The memory 104 may comprise volatile memory and/or non-volatile memory, such as random access memory (RAM), read only memory (ROM), and a hard drive.

[0012] The application 112 may be any type of web-based application that is capable of being hosted by a web-server, such as Apache®, Tomcat®, and Internet Information Server (IIS®). More specifically, the application 112 may comprise one or more software functions and/or data files (not specifically shown) that may utilize web-based technologies, such as Perl, Java®, active server pages (ASP), hypertext preprocessing (PHP), and/or hypertext markup language (HTML), to generate static and/or dynamic content. The methods and systems described herein relate to one or more web-based applications that are operated by a user on the same computer system that is executing the applications.

[0013] The user may interact with the applications without utilizing network protocols, such as TCP/IP and internet packet exchange (IPX), and without opening network ports. The facade server 110 may interact with the application 112 by creating one or more interfaces (not shown in Figure 1), such as application programming interface (APIs) and common gateway interfaces (CGIs), that may ordinarily be created by a web-server. The interfaces allow the application 112 to communicate with the facade server 110 in a substantially similar manner as the application 112 would communicate with a web-server.

[0014] However, the facade server 110 preferably does not use network protocols to communicate with a user of the application 112. Instead, the plugin 108 may exchange application data between the facade server 110 and the browser 106 through a local protocol that is supported by the system 100. The local protocol may facilitate the exchange of data via one or more software component models, such as Component Object Model (COM), named data pipes, memory mapped I/O streams, data files, and other methods of locally transferring data between software components.

[0015] The browser 106 may be capable of rendering application data generated by the application 112 onto a display 114 via an input/output (I/O) interface 116. The browser 106 may be any type of web-browser, such as Internet Explorer®, Netscape®, and Mozilla®. The plugin 108 may be a web-browser plugin, such as a dynamic link library (DLL) using the asynchronous pluggable protocol (APP), a Java® applet, or any other type of plugin that may interact with the browser 106. Other components, such as a keyboard and pointing device (not specifically shown), may be included in system 100 as desired.

[0016] Referring now to Figure 2, an exemplary data flow between the browser 106, the plugin 108, the facade server 110, and the application 112 of Figure 1 is shown. While being executed by the CPU 102 (Figure 1), the facade server 110 may send one or more requests 202 to the application 112. Each request 202 may include an input to the one or more software functions associated with the application 112. Upon receiving a request 202, the application 112 may service the request 202 by executing the appropriate software functions on the CPU 102 (Figure 1) and may generate a corresponding reply 204. The request 202 and corresponding reply 204 are transferred through one or more web-server interfaces 206.

[0017] The facade server 110 may be configured to perform the functions associated with a web-server, such as serving web-pages, parsing data files, and processing server side includes (SSI), without utilizing network protocols, such as TCP/IP, to transfer application data. Instead, the plugin 108 may interact with the facade server 110 via an application programming interface (API) 208. The API 208 may comprise one or more software functions that facilitate the exchange of application data between the plugin 108 and the facade server 110.

[0018] To integrate the plugin 108 with the web browser 106, a local protocol 210 may be registered on the system 100 (Figure 1). Registering a protocol integrates the protocol scheme and one or more protocol handlers with the browser 106. For example, many external protocols, such as file transfer protocol (FTP) and hypertext transfer protocol (HTTP), may be registered with a browser. The registration of the protocols allow the browser to handle protocol-specific schemes, such as FTP and HTTP. The handler of the protocol may be a program responsible for handling references using the protocol scheme. For example, an FTP reference, *e.g.,* ftp://microsoft.com, may be handled by an FTP dynamic link library (DLL) that attempts to establish an FTP connection with a server, *e.g.,* microsoft.com.

[0019] In accordance with embodiments of the invention, the local protocol 210, having a predetermined scheme, may be registered with the web browser 106 and may designate the plugin 108 as the handler of the predetermined scheme. The web browser 106 may submit a request 212 to the plugin 108 by referencing the registered scheme of the local protocol 210.

[0020] The request 212 may include several types of data. The request 212 may include form data that is generated from web-based forms that are rendered by the browser 106, universal resource indicator (URI) data that is generated when a user selects a hyperlink that uses the scheme of the local protocol, or data generated from any other means for exchanging information between a browser and an application through a local protocol. A corresponding reply 214 to the request 212 may be sent from the plugin 108 after the request 212 is processed by the facade server 110 and, if necessary, the application 112.

[0021] In alternative embodiments, the plugin 108, the facade server 110, and the application 112 may be combined into a single component that is executed by the CPU 108 when the local protocol 210 is referenced by the web browser 106. The plugin 108, the facade server 110, and the application 112 may remain distinct, utilizing different address spaces, or be combined to use a common address space.

[0022] In other embodiments, the web server interface 206 may comprise a plurality of interfaces that mimic various types of web-servers, such as Tomcat®, Apache®, and IIS®. To an application being hosted by the facade server 110, the web-server interface 206 may interact with the application in a substantially similar manner as the application would interact with the web-server being mimicked. Thus, multiple protocol handlers may be registered in the system 100, one for each type of web-server being mimicked by the web server interface 206.

[0023] In alternative embodiments, a web-server may be employed in conjunction with the facade server 110 to host one or more applications in a local-only mode and a network mode. Figure 3 illustrates a computer system constructed in accordance with these embodiments of the invention. System 300 may be any type of computer system, such as a laptop computer, a personal computer, or a stand-alone computer operated as a server. The system may comprise a single central processing unit (CPU) 302, as illustrated in Figure 3, or may comprise a plurality of CPUs arranged in a configuration where parallel computing may take place. The CPU 302 may couple to a memory 304 that may store a browser 306 program, a plugin 308 program, a facade server 310 program, a web-server 312 program, and an application 314. In addition, a control file 316 that controls the operational mode of the system 300 is stored in the memory 304. The memory 304 may comprise volatile memory and/or non-volatile memory, such as random access memory (RAM), read only memory (ROM), and a hard drive.

[0024] While operating in the local-only mode, the facade server 310 may handle local requests from the plugin 308, as previously described. While operating in the network mode, the web-server 312 may handle local and remote requests. The mode of operation may be determined by a setting a flag in the control file 316. The browser 306 may be capable of rendering application data generated by the application 314 onto a display 318 via an input/output (I/O) interface 320.

[0025] The above discussion is meant to be illustrative of the principles and various embodiments of the present invention. Numerous variations and modifications will become apparent to those skilled in the art once the above disclosure is fully appreciated. It is intended that the following claims be interpreted to embrace all such variations and modifications.